Page Ranks

Yufei Tao

KAIST

March 17, 2013

Part I

The materials in this part are in the scope of quizzes and exams.

A good library catalog system would offer the following useful search functionality on its book collection: a user inputs a few query terms (e.g., "web search and text analysis"); the system returns the books that are most relevant. We already know how to implement this functionality effectively – this is exactly the relevancy problem we dealt with previously.

There is vast similarity between a library and WWW. If we look at each webpage as a "book", then WWW appears no more than just a collection of books. It thus seems easy to build a search engine – why not just implement the aforementioned search functionality of a catalog system on our WWW "book collection"? Indeed, this was exactly the rationale of some early search engines.

However, Google came up with a new idea to substantially improve the quality of its search engine. After all, WWW is not just a book collection, because books do not have hyperlinks to each other. These hyperlinks, as we will see, provide crucial information that should not be ignored, when looking for the most "influential" webpages among all those deemed relevant to a user's query.

The main motivation of Google is that webpages have different authorities. For example, to many people, a politics article posted on the official page of the US government carries more weight than an article written by an amateur individual. Provided that both articles are identically relevant to a user's query, which one should a search engine recommend to the user? Google believes that it should be the one from the white house.

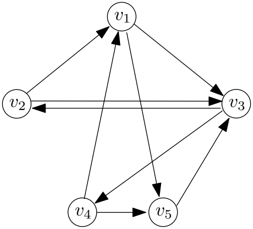
Previously, in solving the term relevancy problem, we calculated the score of a webpage as its text relevance to a query. By the authority argument in the last slide, we know that the score combine both the text relevance and its authority. We will come back to the issue of how to do the combination later. Now, we will discuss how to compute the authority of a webpage.

Graph Modeling of WWW

From now on, we will model WWW as a directed graph $G = (V, E)$. Each webpage is represented as a node in V. Given two nodes (a.k.a. webpages) $v_1, v_2 \in V$, there is a link from v_1 to v_2 in E if there is a hyperlink in webpage v_1 to webpage v_2.

Assumption: To simplify our discussion, we will assume that every node in G has at least one outgoing link.

Here is an example from our reference book:



Let us imagine the following process that mimics the behavior of a user surfing randomly in WWW:

1. Let u be the webpage that the user is currently at.
2. With probability α:
   2.1 Click on a random hyperlink in u.
   2.2 Set u to the new webpage that opens up.
   2.3 Repeat from Step 1.
3. With probability $1 - \alpha$:
   3.1 Set u to a random webpage in WWW – we will refer to this as re-seeding.
   3.2 Repeat from Step 1.

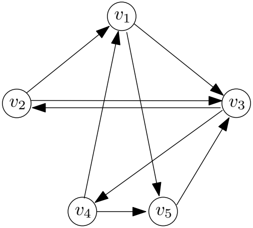
We refer to the above process as Google's random surfing.

Page Rank

Definition (Page Rank)

The authority (a.k.a. page rank) of a webpage equals the probability that it is the t-th webpage visited by the user when t tends to ∞.

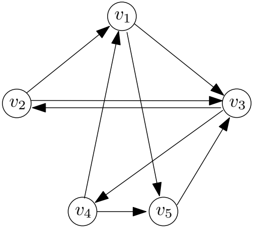
- α is often set to 0.85 in practice.
- To start the process, the first page visited by the user can be any webpage in WWW – its choice does *not* affect the page ranks.



Example: Assume that the first webpage chosen by the user is v_1. Let us analyze the probability that the second page is v_3. For this to happen, one of the following disjoint events must take place:

- Re-seeding happens in choosing the first webpage, and picks v_4. The probability for this is $0.15 \cdot (1/5) = 0.03$.
- Re-seeding does not happen, and the user follows the link from v_1 to v_3. The probability for this is $0.85 \cdot (1/2) = 0.425$.

Hence, the probability for v_3 to be the second webpage is $0.03 + 0.425 = 0.455$.



Example: Let us analyze the probability that the third webpage is v_4. For this to happen, one of the following disjoint events must take place:

- Re-seeding happens in choosing the second page, and picks v_4. The probability for this is $0.15 \cdot (1/5) = 0.03$.
- v_3 is at the second page, re-seeding does not happen, and the user follows the link from v_3 to v_4. The probability for this is $0.455 \cdot 0.85 \cdot (1/2) = 0.193$.

Hence, the probability for v_4 to be the third webpage is $0.03 + 0.193 = 0.223$.

Given a vertex $v \in V$, let $p(v, t)$ be the probability that v is the t-th webpage visited. Then, we have the following recurrence from the above discussion:

$$p(v, t+1) \quad = \quad \frac{1-\alpha}{|V|} + \alpha \cdot \sum_{u \in in(v)} \frac{p(u, t)}{outdeg(u)}$$

where

- $in(v)$ is the set of in-neighbors of v (i.e., nodes with links pointing to v).
- $outdeg(v)$ is the out-degree of v (i.e., the number of out-going links of v).

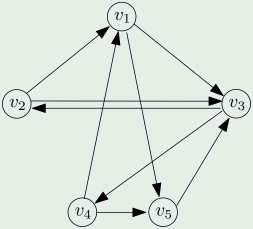
It is guaranteed that, when t is sufficient large:

$$p(v, t+1) = p(v, t)$$

holds for all $v \in V$.

The value of $p(v, t)$ at this moment is referred to as the page rank of v.

Example



Example: The page ranks of $v_1, ..., v_5$ are 0.1716, 0.1666, 0.3214, 0.1666, and 0.1737, respectively. You can write a program to verify this (see the next slide). Later, in Part II, we will learn an easier way to do so.

Page Rank Computation

The following algorithm is called the power method:

1. Let v be an arbitrary node in V. Set $p(v, 1) = 1$, and $p(u, 1) = 0$ for all vertices $u \neq v$.
2. $t = 1$.
3. Use the equation of the previous slide to calculate $p(v, t + 1)$ for all $v \in V$.
4. If $p(v, t) = p(v, t + 1)$ for all $v \in V$, terminate the algorithm.
5. Otherwise, $t \leftarrow t + 1$, and repeat from Step 3.

In practice, Step 4 is usually replaced by "if t is large enough (e.g., $t = 100$), terminate the algorithm".

Document Ranking (Revisited)

Let us revisit the scenario where Google needs to rank the webpages in response to a user's query. Suppose that the query is a sequence Q of terms. In the relevancy problem, our solution was to calculate a relevance score $score(D, Q)$ for each webpage D, and then, rank all the webpages by their scores.

Google, on the other hand, takes into account both the relevance $score(D, Q)$ of D, and its page rank, denoted as $pageR(D)$. Specifically, it calculates a function $f(D, q)$ which monotonically increases whenever $score(D, Q)$ or $pageR(D)$ increases. Then, all documents are ranked in descending order of their values of $f(D, q)$.

The details of $f(D, q)$ have been kept as a commercial secret, on which Google has been granted a patent.

Part II

The following materials are intended for advanced understanding.
They will not be tested in quizzes and exams.

We will discuss how page ranks relate to the well-established theory of Markov chains. In particular, we will see that page ranks form an eigenvector of a matrix that depends on the WWW graph G and α.

Definition (Stochastic Matrix)

An $n \times n$ matrix M is called a stochastic matrix if all the following hold:

- Every value in M is non-negative.
- The values of every row sum up to 1.

From now on, define $M[i, j]$ as the value at the i-th row, and the j-th column of M.

Every stochastic matrix M defines a "random walk" process, formally known as a Markov chain.

- Consider that we have a directed graph G_{mark} of n nodes: $v_1, ..., v_n$. For every non-zero entry $M[i, j]$ of M ($1 \leq i, j \leq n$), G_{mark} has an edge from v_i to v_j (note: j can be i, namely, there can be self-loop edges).
- At the beginning of the random walk, you stand at any vertex of your choice – this is your first stop.
- Then, inductively, assuming you are at a node v_i at the t-th stop ($t \geq 1$), you move to a neighbor v_j with probability $M[i, j]$. The new node you are standing at now is the $(t + 1)$-th stop.

Definition (Irreducibility)

An $n \times n$ stochastic matrix M is irreducible if, for all $1 \leq i, j \leq n$, there is a path from v_i to v_j in G_{mark}.

Definition (Probability Vector)

An $n \times 1$ vector P is a probability vector if both the following are true:

- Each component in P is a value between 0 and 1.
- All components of P sum up to 1.

Theorem

Let M be an irreducible stochastic matrix corresponding to Google's random walk, and M^T be the transpose of M. The following statements are correct:

- There is a unique probability vector P satisfying $P = M^T P$.
- M^T has a an eigenvalue 1. All the other eigenvalues of M^T have absolute values strictly less than 1.

The process of Google's random surfing can be regarded as a Markov chain. Specifically, assume that WWW has n webpages $v_1, ..., v_n$. If you are currently at webpage v_i, then you jump to webpage v_j as the next stop with probability:

- $\frac{1-\alpha}{n}$, if v_i does not have a hyperlink to v_j.
- $\frac{1-\alpha}{n} + \frac{\alpha}{outdeg(v_i)}$, if v_i has $outdeg(v_i)$ hyperlinks, one of which points to v_j.

You can view the above process as a Markov chain on a graph G_{mark}, where each v_i corresponds to a webpage, and there is a link from every v_i to every v_j (even for $i = j$). Let M be the matrix for this Markov chain. Then, $M[i, j]$ is set as the probability of jumping from v_i to v_j as discussed above.

Think

Verify by yourself that M is an irreducible stochastic matrix.

As before, let $p(v_i, t)$ $(1 \le i \le n)$ be the probability that webpage v_i is the t-th one visited by the random surfer. Let $P(t)$ be an $n \times 1$ vector such that:

$$P(t) = (p(v_1, t), p(v_2, t), ..., p(v_n, t))^T$$

where the superscript T stands for "transpose".

From Slide 13, we know:

$$P(t+1) = M^T \cdot P(t).$$

When $P(t+1) = P(t)$, the values in $P(t)$ give the page ranks of the vertices $v_1, ..., v_n$. At this moment, $P(t)$ is the solution of P from the following equation:

$$P = M^T \cdot P.$$

Namely, P (which is a probabilistic vector) is an eigenvector of M of eigenvalue 1. By the theorem in Slide 22, P exists and is unique.

Remark: For this reason, P is commonly referred to as the stationary probability vector of the Markov chain described by M.

With everything said, we can now re-state the power method in a concise manner:

1. Set $P(1) \leftarrow (1, 0, ..., 0)^T$, and $t \leftarrow 1$.
2. Compute

$$P(t+1) \quad = \quad M^T \cdot P(t).$$

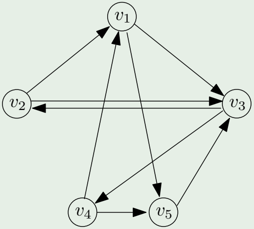
3. $t \leftarrow t + 1$.
4. Repeat from Step 2.

Theorem

Let M be an irreducible stochastic matrix corresponding to Google's random surfing, and P be the stationary probability vector of the Markov chain described by M. Then, in the power method, $\lim_{t\to\infty} P(t) = P$.

In practice, the value α controls the convergence rate, i.e., how far $P(t)$ gets close to P. In particular, the smaller is α, the faster the convergence.

Example



The matrix describing the random walk is:

$$M = \begin{bmatrix} 0.03 & 0.03 & 0.455 & 0.03 & 0.455 \\ 0.455 & 0.03 & 0.455 & 0.03 & 0.03 \\ 0.03 & 0.455 & 0.03 & 0.455 & 0.03 \\ 0.455 & 0.03 & 0.03 & 0.03 & 0.455 \\ 0.03 & 0.03 & 0.88 & 0.03 & 0.03 \end{bmatrix}$$

You can verify that $P = (0.1716, 0.1666, 0.3214, 0.1666, 0.1737)^T$ is an eigenvector of M^T with eigenvalue 1. It is the stationary probability vector of the Markov chain described by M.